## REMARKS

Reconsideration and allowance of the above referenced application are respectfully requested.

Upon entry of this amendment, claims 1, 3-5, 8-9, 11-14, 16-18, 21-22, 24-27, 29-31, 34-35, and 37-39 will remain in the application.

Claims Rejections - 35 USC § 103

Claims 1-3, 9, 12-16, 22, 25-29, 35, 38 and 39 were rejected under 35 USC 103(a) for allegedly being unpatentable over Sheard et al. (US 6,208,345, hereinafter "Sheard")

Claims 4-8, 10, 11, 17-21, 23, 24, 30-34, and 36-37 were rejected under 35 USC 103(a) for allegedly being unpatentable over Sheard, and further in view of Amado (US 5,701,400).

The invention formalizes the parameterization of data flow graphs to allow runtime parameters. Runtime parameters allow an application builder to defer the value of a parameter setting (e.g., the key parameter of a sort function, file names, record formats, transform functions, etc.) to runtime (i.e., the time an application is executed on a computer system). The values of runtime parameters may be supplied by the end user or be derived from a combination of other runtime parameters or objects stored in an object repository or be determined from a default setting.

Runtime parameters add a certain amount of flexibility to an application. Additional flexibility is achieved by using those parameters to compute metadata (data formats or types, and program logic or transforms) on demand. Types and transforms may be synthesized from other types and transforms, user-supplied

parameter values, and stored objects (e.g., from a repository). This makes it possible to build "generic" applications that work on input data of any type, or that produce data through a series of transforms whose construction is controlled, directly or indirectly, through runtime parameter values.

One embodiment of the invention includes a conditional components mechanism that permits changes to a graph structure based on parameter values and computed metadata. Each component of a graph has a condition which controls whether or not that component will appear in the graph at runtime. The condition can be computed directly or indirectly through runtime parameters. Conditional components can be used to optimize or specialize graphs.

The principal reference cited by the Examiner, Sheard, alone or in combination, fails to teach or suggest the invention as claimed.

While Sheard teaches a visual interface for building a transport framework which facilitates the exchange of technology-dependent data between disparate applications, it does not address the technology of graphs having runtime parameters or of modifying graphs.

With respect to independent claims 1, 14, and 27, as amended, and their dependencies, Sheard fails to teach or suggest executing a graph as claimed. The Examiner cites FIG. 21 as illustrating the step of retrieving a runtime parameter, when at best it teaches (in conjunction with explanatory text at col. 24, ll. 28-50) double-clicking on an icon to display statistical charts or configuration settings, or right clicking on a queue to display its contents. This manual action is not the same as

programmatically retrieving a runtime parameter for a graph at runtime execution of the graph. The text cited in Col. 3, ll. 44-50 as teaching determining whether the value of a runtime parameter is to be provided by user input neither mentions or suggests such a concept, but simply describes a visual interface for runtime control and analysis of the business and technical aspects of an integrated information system deployment, and that visual views of the runtime system deployment provide consistent management and control for a variety of users having different data input/output, reporting, and interface requirements. Sheard at col. 24, ll. 45-50 cannot be fairly said to teach executing a graph using a final parameter value since Sheard does not discuss graphs (as defined by Applicants) at all.

However, to make clearer the distinctions of the present invention over Sheard, Applicants have amended claims 1, 14, and 27 to recite that a final parameter value is based on one of (a) user input to a prompt, (b) an externally supplied value, or (c) a default value, where cases (a) and (b) respectively occur upon a determination that the value for a runtime parameter is to be provided by user input or is to be externally supplied programmatically. Nothing in Sheard, alone or in combination, teaches or suggests such characteristics. Accordingly, Applicants submit that claims 1, 14, and 27, and their dependences, are allowable.

With respect to independent claims 9, 12, 22, 25, 35, and 38, as amended, and their dependencies, Sheard fails to teach or suggest modifying a graph as claimed. The portion of Sheard referenced by the Examiner as teaching modifying a graph in fact simply shows two different views of the same application: FIG.

18 is a System Integration View selected by clicking on tab 520 (see FIG. 17) while FIG. 20 is a Business Integration View of the same thing, selected by clicking on tab 526 (Col. 20, ll. 3-30); the only difference between the views is that a statistical analysis icon is made visible in FIG. 20. FIG. 21 has nothing to do with determining conditional components, but instead shows a queue status chart developed from the data integration implementation shown in FIG. 18. The text cited in Col. 3, ll. 44-50 as teaching evaluating a condition mentions nothing of the sort (and is the same text that the Examiner asserts teaches determining whether the value of a runtime parameter is to be provided by user input - an entirely different concept from evaluating a condition for a conditional component; these 6 lines of text in Sheard teach neither concept). Lastly, the text referenced in Sheard as teaching modifying a graph (col. 24, ll. 45-50) simply teaches that double-clicking on a component display displays charts of information trends - there is no structural change to a graph.

In contrast, claims 9, 22, and 35, as amended, require modifying a graph by removing at least one conditional component from the graph. Claims 12, 25, and 38, as amended, require modifying a graph by replacing at least one conditional component with a flow before execution of the graph. These are structural changes to the application represented by the graph, neither of which are taught or suggested by Sheard. In both claims, the modified graph is then executed. Nothing in Sheard, alone or in combination, teaches or suggests such modification of executable graphs. Accordingly, Applicants submit that

claims 9, 12, 22, 25, 35, and 38, and their dependences, are allowable.

Enclosed is a $274 check for excess claim fees and a $475 check for the Petition for Extension of Time fee. Please apply any other charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: 3/12/04

Kenyon S. Jenckes
Reg. No. 41,873

Fish & Richardson P.C.
PTO Customer No. 20985
12390 El Camino Real
San Diego, California 92130
Telephone: (858) 678-5070
Facsimile: (858) 678-5099

10374968.doc